In the Specification:

On page 1, immediately following the RELATED APPLICATION INFORMATION heading, please replace Paragraph **[0001]** with the text as shown:

[0001] This application ~~is a continuation-in-part of~~ claims the benefit of and priority to provisional Application No. 60/246,752, filed 9 November 2000, entitled *Methods and Systems for Categorizing Information*, by these same inventors Clark Breyman and Mark Davis, which application is hereby incorporated by reference.

On page 1, please replace Paragraph **[0002]** with the text as shown:

[0002] Training sets are used in automatic categorization of documents, to establish precision and recall curves and to train automatic categorization engines to ~~categorized~~ categorize documents correctly. Precision and recall curves are standard measures of effective categorization and information retrieval. Precision is a measure of the proportion of documents retrieved that are relevant to the intended result. Recall is a measure of the coverage of a query, for instance the number of documents retrieved that match an intended result, compared to the number of documents available that match the intended result. Precision and recall curves generated from a training set are used to set criteria for automatically assigning documents to categories. The training set typically includes documents with categories that have been editorially established or verified by a human.

On page 4, please replace Paragraph **[0015]** with the text as shown:

[0015] Figure 3 depicts a pair of precision and recall curves. Precision is standard measure of information retrieval performance. It is defined as the number of relevant documents retrieved divided by the total number of documents retrieved. For example, suppose that there are 80 documents relevant to widgets in the collection. A retrieval system returns 60 documents, 40 of which are about widgets. The system's precision is 40/60 = 67 percent. In an ideal world, precision is 100 percent. Since this is easy to achieve (by returning just one document,) the system attempts to maximize both precision and recall simultaneously. Recall, again, is defined as the number of relevant documents retrieved divided by the total number of relevant documents in the collection. For example, suppose that there are 80 documents relevant to widgets in the collection. The system returns 60 documents, 40 of which are about widgets. Then the system's recall is 40/80 = 50 percent. In an ideal world, recall is 100 percent. Like perfect precision, perfect recall is trivial to achieve (by retrieving all of the documents.) Thus, it is important to measure system performance by both precision and recall. One standard way of plotting precision and recall curves is to determine thresholds that recall 0, 10, 20 .... 100 percent of the relevant documents in the collection. The recall curve 302 is plotted at such varying degrees of recall, expressed as a percentage 312. At these thresholds for recall, the precision score 311 is also calculated, expressed as a fraction 311. A ~~recall~~ precision curve 301 is juxtaposed on the same graph. This pair of curves illustrates that as recall increases, precision tends to drop. The two are inversely related, but not precisely related. The choice of appropriate parameters or thresholds to trade-off precision and recall depends on the shape of precision and recall curves for a particular topic and the preferences of the user community.

On page 4 and continuing onto page 5, please replace Paragraph **[0017]** with the text as shown:

[0017] A data source 490, a data sink 492 and an input device 494 may be connected to the data organization system 400 by links 491, ~~492 and 493,~~ **493 and 495,** respectively. The data source 490 can be a locally or remotely located computer or database, or any other known or later developed device that is capable of generating electronic data. Similarly, the data source 490 can be any suitable device that stores and/or transmits electronic data, such as a client or a server of a network. The data source 490 can alternatively be a storage medium such as a magnetic or optical disk that is accessed by a suitable device located within or externally of the data organization system 400. The data source 490 can be connected to the data organization system 400 over a connection device, such as a modem, a local area network, a wide area network, an intranet, the Internet, any other distributed processing network, or any other known or later-developed connection device.

On page 9 and continuing onto page 10, please replace Paragraph **[0037]** with the text as shown:

[0037] An exemplary method for implementing the invention is given as follows:

```
// ~~Pseduocode~~ Pseudocode for Training Set Analysis System
//
// This ~~pseduocode~~ pseudocode is intended to exemplify one possible
// implementation of the training set analysis system.
//
// Syntax and ~~sematics~~ semantics of this pseudocode roughly approximate
// that of C++.
//

void ~~AnalyizeTrainingSet~~ AnalyzeTrainingSet(TrainingSet originalTS) {
 // Iterate over members of the original training set
 foreach (TrainingSample S in originalTS) {
  // Identify the test sample and create a test training set
  // without the test sample.
  TrainingSets    testTS;

  testTS.CopyFrom(originalTS);
  testTS.RemoveSample(S);

  // Train a categorization by a example system with the
  // test training set
  CategorizationSystem testCategorizer;
  testCategorizer.Train(testTS);

  // Categorize the test sample with the testCategorizer and
  // compare the automatic categorization results with the original
  // categories.
  Categories  originalCategories = S.GetCategories();
  Categories  testCategories     = testCategorizer.Categorize(S.Data());

  foreach (Category C in originalCategories) {
   if (C not in testCategories) {
      printf("Potentially incorrect category %s on sample %s\n",
        C.Name(), S.Name());
   }
```

```
    }

    foreach (Category C in testCategories) {
      if (C not in originalCategories) {
          printf("Potentially missing category %s on sample %s\n",
             C.Name(), S.Name());
      }
    }
  }
}
```

On page 10 and continuing onto page 11, please replace Paragraph **[0038]** with the text as shown:

[0038] In one exemplary implementation, a program may be invoked from a command line using the syntax: tsanalysis <lxcatoptions> <minPPresent> <maxPMissing> <output file>. In this implementation, the program accepts four input parameters: ***lxcatoptions:*** the categorization model configuration file (lxcatoptions). ***minPPresent:*** The minimum ~~probablity~~ probability of correctness for codes in the training set. The program records as *suspicious* training document topic codes that fails to meet this threshold. For example, if the minPPresent value is 0.01, the program will record as ~~suspiceous~~ suspicious topic code instances with less than a 1% chance of being correct. This parameter must be greater than 0. ***maxPMissing:*** The maximum ~~probablity~~ probability of correctness for codes absent from the training set. The Analysis Tool records as *missing* training document topic codes that exceeds this threshold but has not been applied to the document. For example, if the maxPMissing value is 0.99, the Accuracy Tool will record as missing topic code instances not applied to a training document despite a 99% likelihood of being correct. This parameter must be greater than 0. ***output file:*** The file name for the Analysis Tool's report. A pair of sample outputs appear above.

//

//

On page 14, please replace Paragraph **[0047]** with the text as shown:

[0047] The nearest neighbors to the probe document have one or more topic assignments T_i. The category scores of the topics can then be calculated for the probe document in at least two ways:

$$\Omega_0(\mathbf{d}_t, T_m) = \sum_{\mathbf{d} \in \{K(\mathbf{d}_t) \cap T_m\}} s(\mathbf{d}_t, \mathbf{d}) \qquad (0.1)$$

or:

$$\Omega_1(\mathbf{d}_t, T_m) = \frac{\sum_{\mathbf{d}_1 \in \{K(\mathbf{d}_t) \cap T_m\}} s(\mathbf{d}_t, \mathbf{d}_1)}{\sum_{\mathbf{d}_2 \in K(\mathbf{d}_t)} s(\mathbf{d}_t, \mathbf{d}_2)} \qquad (0.2)$$

The first formula calculates the sum Ω_1 of the contributions or evidence that probe document $\mathbf{d}_t$ should receive topic assignment T_m, while the second calculates the normalized sum Ω_2, normalized to reflect the density of the neighborhood of the probe document. These formulas may be modified to reflect the number of documents among the nearest neighbors that share the topic assignment T_m. Figures 5A-B illustrate the significance of normalizing scores. An ambiguous situation, as in Figure 5A, may appear to strongly support a topic assignment, because the neighborhood of the test document is dense. An unambiguous situation, as in Figure 5B, may seem to ~~weekly~~ weakly support a topic assignment, unless a normalized metric of confidence is used, because the neighborhood to the test document is less dense.